

# **DEV290**

## **Exploring What's New In The CLR 2.0**

**Kit George**

**([kitg@microsoft.com](mailto:kitg@microsoft.com))**

**Program Manager**

**Common Language Runtime**

**Microsoft Corporation**

# Developer Roadmap

## Visual Studio .NET 2003

- “Everett Release”
- Windows Server 2003 integration
- Support for .NET Compact Framework and device development
- Improved performance

## Visual Studio 2005 “Yukon”

- “Whidbey” release
- SQL Server integration
- Improved IDE productivity and community support
- Extended support for XML Web services

## Visual Studio Orcas

### “Longhorn”

- “Orcas” release
- Windows “Longhorn” integration
- New UI tools and designers
- Extensive managed interfaces

# Agenda

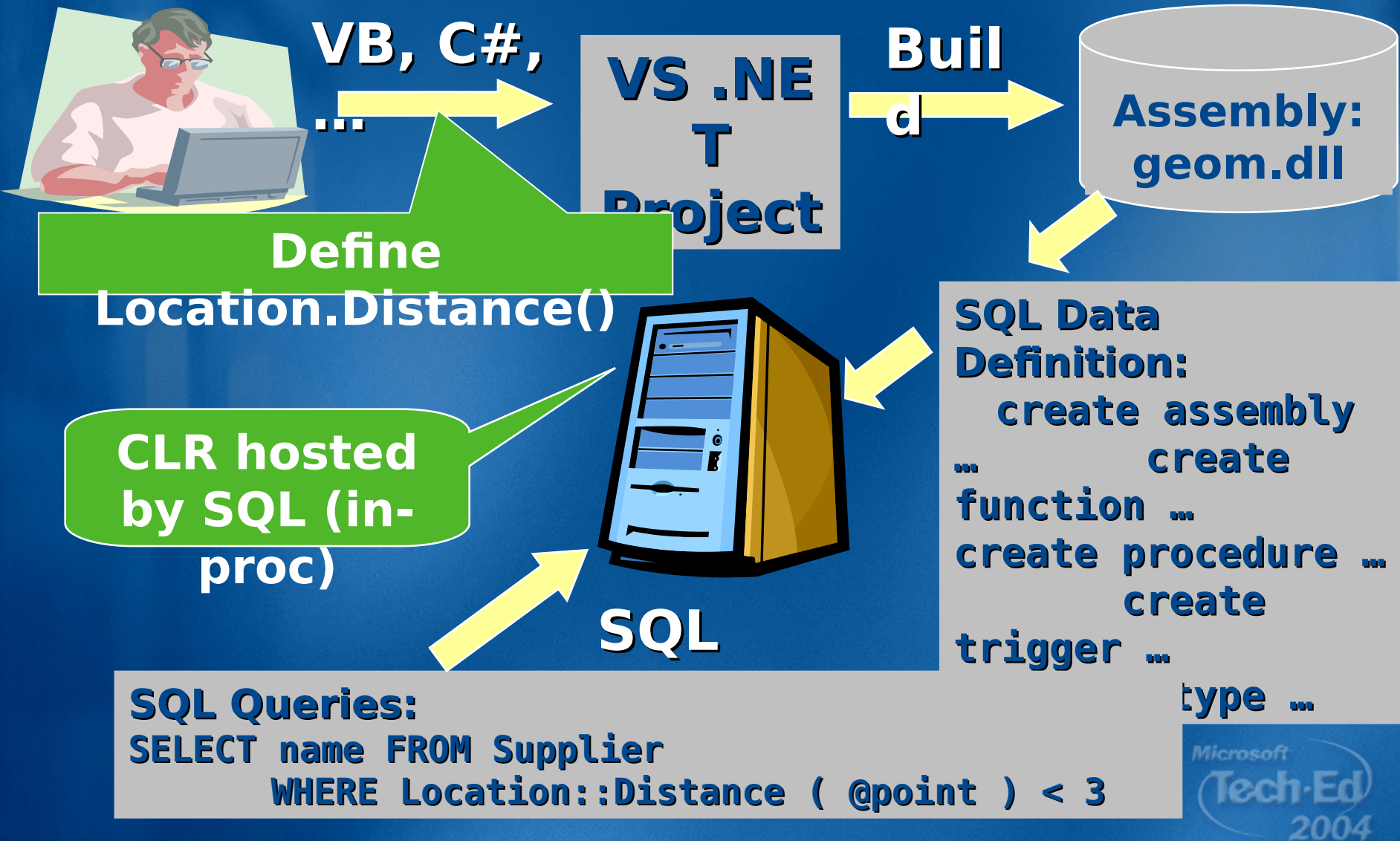
- **Extending the Platform**
  - SQL Server Integration
  - Moving to 64 bit
- **Improving the Platform**
  - Performance work
  - RAD Debugging
  - Security improvements
- **Innovating in the Platform**
  - Generics support
  - BCL Enhancements



# SQL Server Integration Design Goal

- Bring framework programming model into the database tier
- Allow business logic to easily migrate to the most appropriate tier
- Enable *safe* database extensions
- **Result:** Stored Procedures, Triggers, data types defined in

# SQL CLR Functionality



# T-SQL

# Splitting a string

```

declare @str varchar(200)
select @Str = 'Microsoft Corporation|SQL
SQL-CLR|2002-08-20|11:32:00|Document|3.
SELECT
substring(@Str + '|', 0 + 1,
    charindex('|', @Str + '|', 0 + 1) - 0)
substring(@Str + '|', charindex('|', @S
    charindex('|', @Str + '|',
    charindex('|', @Str + '|') + 1) -
    charindex('|', @Str + '|') - 1 ),
substring(@Str + '|', charindex('|', @S
    charindex('|', @Str + '|') + 1) + 1,
    charindex('|', @Str + '|', charindex
'|',
    charindex('|', @Str + '|') + 1) + 1)
    charindex('|', @Str + '|',
    charindex('|', @Str + '|') + 1) - 1 )
substring(@Str + '|', charindex('|', @S
    charindex('|', @Str + '|')

```

[illegible]

# Sql Programming Model



## ● Splitting a string The new way....

```
Public Shared Sub SplitString()  
    Dim s As String  
    s = "Microsoft Corporation|SQL Server|
```

### Other Talks

**Dat318 - SQL Server 2005: .NET Framework-  
Based Programming in the Database**

**Wednesday, 3:45**

**Room 5AB**

**DEV394 - Migrating to Managed Code**

**Friday, 2:45**

**Room 20D**



# demo

## **Strongly Typed Resources**



**The category is 64 Bit**  
**The answer for \$2000**  
**is...**

**YES!**

# Moving to 64 bit

- **64 bit for Servers and workstations**
  - **X64 and IA64 bit support**
  - **Enable SQL Server 2005 and ASP.NET**
- **Verifiable managed binaries just run!**
- **Support for WoW64**
  - **Enables machine**
  - **1.0 and WoW**

**Talks already presented**

**Dev332 - .NET Framework: 64-Bit Development Today and Tomorrow**

# demo

## **SerialPort Communication**



# Performance

Objectives: make .NET an even greater programming platform

- **Long-Term: make the performance characteristics of the CLR similar to native code**
- **Reduce marginal cost of additional managed processes**
- **Reduce startup time and working set**

## NGen

- **Compiles IL code to native code, saving results to disk**
- **Advantages: no need to recompile IL to native code, and class layout already set so better startup time**
- **VS 2005: Significant reductions in the amount**

# Performance

## New, Performant APIs everywhere

- APIs for faster resource lookup
- Lightweight CodeGen: only generates essential code (contrast to Reflect Emit)

## Existing APIs

- C  
Be  
si  
**Other Talks**  
**Dev462 - Windows Forms: Tips and Tricks To Improve Your Performance**  
**Wednesday, 8:30** **Room 2**
- D  
m
- A  
re  
**Dev491 - .NET Framework: Writing Faster Managed Code**  
**Thursday, 1:30** **Room 20A**

# demo

## **TryParse and MemoryPressure**



# **RAD Debugging**

- **Edit and Continue: Edit Code at runtime**

## **Allowed Edits: Examples**

- Add private fields to a class
- Add private non-virtual methods to a class
- Change a function body, even while stepping

## **Disallowed Edits: Examples**

- Removing fields/methods
- Edits to generic classes
- Serialization will not recognize new fields
- **Display Attributes for a better debugging experience**

# demo

## **Debugger Display Views**

# CLR Security

- **New cryptography support**
  - PKCS7 support
  - XML encryption support
  - Enhanced support for X509 certificates
- **Enhanced Application Security**
  - Permission Calculator
    - Integration with ClickOnce
  - Debug-In-Zo
- **Managed ACL**

**Talks Already Presented**  
DevC02 - .NET  
Framework Security:  
Best Practices



# demo

## **IO and other Improvements**

# Generics



```
public class List<T>
{
    private T[] elements;
    private int count;

    public void Add(T element) {
        if (count == elements.Length) Resize(count * 2);
        elements[count++] = element;
    }

    public T this[int index] {
        get { return elements[index]; }
        set { elements[index] = value; }
    }

    public
    }
}
```

```
List<int> intList = new List<int>();
```

```
intList.Add(1);           // No boxing
intList.Add(2);           // No boxing
intList.Add("3");         // Compile-time error
```

```
int i = intList[0];       // No cast required
```

# Generics

- **Why generics?**
  - Compile-time type checking
  - Performance (no boxing, no downcasts)
  - Reduced code bloat (typed collections)
- **VB, C#, MC++ produce & consume generics**
- **Use generics freely in internal APIs**
- **Consider using generics in public APIs**
  - Generics are not yet in CLS
  - To be CLS compliant, provide a non-generic API alternative
- **Microsoft is actively pursuing standardization of generics in runtime and languages**



# Generics in VB



```
Public Class List(Of ItemType)
    Private elements() As ItemType
    Private elementcount As Integer
    Public Sub Add(ByVal element As ItemType)
        If elementcount = elements.Length Then
            Resize(elementcount * 2)
            elements(elementcount) = element
            count += 1
        End Sub
    Public Default Property Item(ByVal index As Integer) As ItemType
        Get
            Return elements(index)
        End Get
        Set (ByVal Value As ItemType)
            elements(index) = Value
        End Set
    End Property
    Public ReadOnly Property Count As Integer
        Get
        End Get
    End Property
End Class
```

```
Dim intList As New List(Of Integer)
intList.Add(1)           ' No boxing
intList.Add(2)           ' No boxing
intList.Add("3")         ' Compile-time error
Dim i As Integer = intList(0) ' No cast required
```

# Generics in C++



```
generic<typename T>
public ref class List {
    array<T>^ elements;
    int count;
public:
    void Add(T element) {
        if (count == elements->Length) Resize(count * 2);
        elements[count++] = element;
    }
    property T default [int index] {
        T get() { return elements[index]; }
        void set(T value) { elements[index] = value; }
    }
    property int Count {
        int get() { return count; }
    }
};
```

```
List<int>^ intList = gcnew List<int>();
intList->Add(1);           // No boxing
intList->Add(2);           // No boxing
intList->Add("3");         // Compile-time error
```

```
int i = intList[0];       // No cast required
```

# Enhancing the Base Library

## • System.Collections.Generic classes

- List<T>
- Dictionary<K, V>
- Stack<T>
- Queue<T>

## • System.Collections.Generic interfaces

- IList<T>
- IDictionary<K, V>
- ICollection<T>
- IEnumerable<T>
- IEnumerator<T>
- IComparable<T>



# Enhancing The Base Library

- **Nullable(Of T)**
  - Other Generics to look out for
    - Extremely useful for situations where null is a value, such as database values

```
Dim intVal as Nullable(Of Integer) = 5  
If intVal.HasValue Then ' checks for a value
```

## Other Talks

• **Dev321 - Visual C# 2005: Language Enhancements**

**Wednesday, 10:15**

**Room**

**31ABC**

deleg  
where

# VS 2005 Tracing

## Additional Listeners

### ● Pre - VS 2005

- Formatting of Trace information was predefined, and not controllable
- No Console listener, ASP had their own tracing mechanism

### ● VS 2005

- Predefined listeners (Console, ETW) and you can make your own
- Integrated ASP Tracing into our own
- Support formatting of output, into defined formats, such as delimited, or XML

# VS 2005 Tracing

## Features

### Simple Thread Identification

- **Pre - VS 2005** : specific threads were difficult to identify, and filter on
- **VS 2005** : Correlation ID has been added, which allows quick and easy identification of a thread, and simple subsequent filtering

### Auto-generated Data

- **Pre - VS 2005** : Standard information such as timestamp or callstack would have to be explicitly generated
- **VS 2005** : An Admin can easily generate this data automatically, via config settings

<listeners>

```
<add name="example.log" TraceOutputOptions=
    "Timestamp, CorrelationId"
```



# VS 2005 Tracing

## Features

### Listener Filtering

- **Pre - VS 2005** : Filtering of Trace information was predefined, and not controllable
- **VS 2005** : Assign filtering to listeners, so only certain messages are traced. This allows filtering on any property of a message. E.g., ID, TimeStamp, etc.

### Switches

- **Pre - VS 2005** : Couldn't determine what tracing a component supported
- **VS 2005** : Support the ability to determine what tracing mechanisms

# demo

## Putting it together...

# Summary

- **Extending the Platform**
  - SQL Server Integration
  - Moving to 64 bit
- **Improving the Platform**
  - Performance work
  - RAD Debugging
  - Security improvements
- **Innovating in the Platform**
  - Generics support
  - BCL Enhancements



# Announcing: CLR Application Compatibility Lab

Visit

<http://www.gotdotnet.com/team/changeinfo/compatibility/> for details

Or

- Simply send your name, company name, and a contact email address to [clrcfb@microsoft.com](mailto:clrcfb@microsoft.com)

You will be sent details, and information about the appcompat toolkit

# Resources

- **BCL Blog**  
<http://weblogs.asp.net/bclteam>
- **BCL Website**  
<http://www.gotdotnet.com/team/clr/bcl/default.aspx>
- **BCL public mail alias**  
[bclpub@microsoft.com](mailto:bclpub@microsoft.com)
- **Reference (Brad Abrams)**  
**.NET Framework Standard Library Annotated Reference**
- **My email: [kita@microsoft.com](mailto:kita@microsoft.com)**

# Other Talks

**Dat318 - SQL Server 2005: .NET Framework-Based  
Programming in the Database**

**Wednesday, 3:45    Room 5AB**

**DEV394 - Migrating to Managed Code**

**Friday, 2:45            Room 20D**

**Dev491 - .NET Framework: Writing Faster Managed Code**

**Thursday, 1:30        Room 20A**

**DevC37 - .NET Framework BCL Tips and Tricks**

**Thursday, 1:30        Cabana 6**

**DevC17 - .NET Framework COM Interop Best Practices**

**Thursday, 3:15        Cabana 5**

**DevC21 - Managed Code Versioning: Patterns and practices  
for increasing the longevity of your managed  
code**

**Friday, 3:15            Cabana 5**



# Community Resources

Attend a free chat or web cast

<http://www.microsoft.com/communities/chats/default.mspx>

<http://www.microsoft.com/usa/webcasts/default.asp>

List of newsgroups

<http://communities2.microsoft.com/communities/newsgroups/en-us/default.aspx>

MS Community Sites

<http://www.microsoft.com/communities/default.mspx>

Locate Local User Groups

<http://www.microsoft.com/communities/usergroups/default.mspx>

Community sites

<http://www.microsoft.com/communities/related/default.mspx>

**Q & A:**  
**We want your feedback!**



# ***Microsoft***<sup>®</sup>

*Your potential. Our passion.*<sup>™</sup>

© 2004 Microsoft Corporation. All rights reserved.

This presentation is for informational purposes only. Microsoft makes no warranties, express or implied, in this summary.

